

**This Page Is Inserted by IFW Operations  
and is not a part of the Official Record**

## **BEST AVAILABLE IMAGES**

**Defective images within this document are accurate representations of the original documents submitted by the applicant.**

**Defects in the images may include (but are not limited to):**

- **BLACK BORDERS**
- **TEXT CUT OFF AT TOP, BOTTOM OR SIDES**
- **FADED TEXT**
- **ILLEGIBLE TEXT**
- **SKEWED/SLANTED IMAGES**
- **COLORED PHOTOS**
- **BLACK OR VERY BLACK AND WHITE DARK PHOTOS**
- **GRAY SCALE DOCUMENTS**

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-334055

(43)Date of publication of application : 18.12.1998

(51)Int.Cl.

G06F 15/16

(21)Application number : 09-145383

(71)Applicant : SONY CORP

(22)Date of filing : 03.06.1997

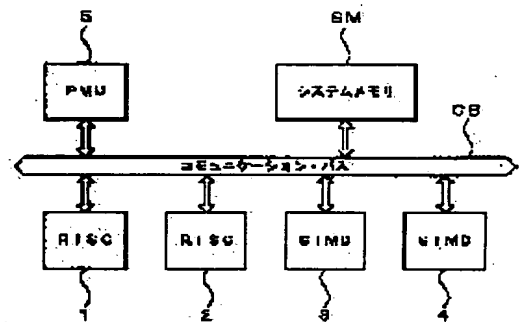
(72)Inventor : TAMURA YUKIHIRO

## (54) MULTIPROCESSOR SYSTEM

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a multiprocessor system to easily add an extension function peculiar to specified application software to a system while holding upward compatibility of the system.

**SOLUTION:** The system is provided with a means to assign a process to be executed in the system to a virtual processor, a means to generate status information whose contents are whether the process assigned to the virtual processor is the process peculiar to the specified application software or not, all purpose physical processors 1, 2, processors 3, 4 for the process peculiar to the specified application software and a means 5 to execute the virtual processor by switching the virtual processor to the all purpose processors 1, 2 and the processors 3, 4 for the proper process according to the contents of status information.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-334055

(43) 公開日 平成10年(1998)12月18日

(51) Int.Cl.<sup>°</sup>

G 0 6 F 15/16

識別記号

3 7 0

F I

G 0 6 F 15/16

3 7 0 N

審査請求 未請求 請求項の数 1 O L (全 7 頁)

(21) 出願番号

特願平9-145383

(22) 出願日

平成9年(1997)6月3日

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 田村 征大

東京都品川区北品川6丁目7番35号 ソニー株式会社内

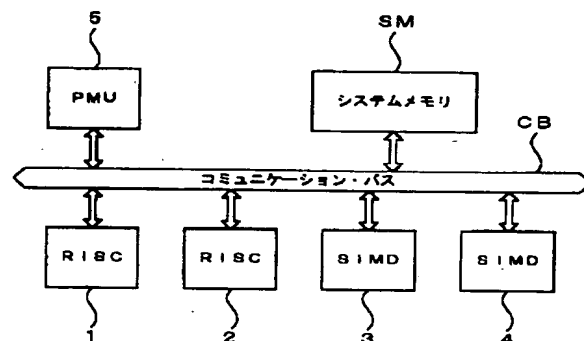
(74) 代理人 弁理士 松隈 秀盛

(54) 【発明の名称】 マルチプロセッサ・システム

(57) 【要約】

【課題】 特定のアプリケーションソフトウェアに固有の拡張機能を、システムの上位互換性を保ったまま容易にシステムに追加できるようにしたマルチプロセッサ・システムを提供する。

【解決手段】 システムにおいて実行すべきプロセスを仮想的なプロセッサに割り当てる手段と、仮想的なプロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスであるか否かを内容とするステータス情報を作成する手段と、汎用物理プロセッサ1、2と、特定のアプリケーションソフトウェアに固有のプロセスのためのプロセッサ3、4と、ステータス情報の内容に応じて、仮想的なプロセッサを汎用プロセッサ1、2と固有のプロセスのためのプロセッサ3、4とに切り替えて実行させる手段5とを備えている。



**【特許請求の範囲】**

**【請求項1】** システムにおいて実行すべきプロセスを仮想的なプロセッサに割り当てる手段と、

前記仮想的なプロセッサに割り当てられた前記プロセスが特定のアプリケーションソフトウェアに固有のプロセスであるか否かを内容とするステータス情報を作成する手段と、

汎用プロセッサと、

前記固有のプロセスのためのプロセッサと、

前記ステータス情報の内容に応じて、前記仮想的なプロセッサを前記汎用プロセッサと前記固有のプロセスのためのプロセッサとに切り替えて実行させる手段とを備えたことを特徴とするマルチプロセッサ・システム。

**【発明の詳細な説明】****【0001】**

**【発明の属する技術分野】** 本発明は、複数のプロセッサを結合し、統一したOS（オペレーティングシステム）のもとでハードウェア資源及びソフトウェア資源を共用した並列処理システムであるマルチプロセッサ・システムに関し、特に、特定のアプリケーションソフトウェアに固有の拡張機能を、システムの上位互換性を保ったまま容易にシステムに追加できるようにしたものに関する。

**【0002】**

**【従来の技術】** 従来、パーソナル・コンピュータ等で採用されているSMP（対称型マルチプロセッサ）システムをはじめとする各種マルチプロセッサ・システムにおいて、例えばマルチメディア処理のような特定のアプリケーションソフトウェアに固有の拡張機能を追加する場合には、システム内の汎用プロセッサに、当該処理のための特別な演算器等を付加するような大幅なアーキテクチャの変更を行っていた。

**【0003】**

**【発明が解決しようとする課題】** このような汎用プロセッサ自体のアーキテクチャの大幅な変更を伴う拡張機能の追加を、システムの上位互換性を保ったまま行うことは、大変困難な作業であった。また、こうした拡張機能の使用頻度は通常はそれほど高くないので、その機能の追加のためにこうした困難な作業を行うことは、非効率的であった。

**【0004】** 本発明は上述の点に鑑みてなされたもので、特定のアプリケーションソフトウェアに固有の拡張機能を、システムの上位互換性を保ったまま容易にシステムに追加できるようにしたマルチプロセッサ・システムを提供しようとするものである。

**【0005】**

**【課題を解決するための手段】** 本発明に係るマルチプロセッサ・システムは、システムにおいて実行すべきプロセスを仮想的なプロセッサ（以下、仮想プロセッサと呼ぶ）に割り当てる手段と、仮想的なプロセッサに割り当

てられた前記プロセスが特定のアプリケーションソフトウェアに固有のプロセスであるか否かを内容とするステータス情報を作成する手段と、汎用プロセッサと、特定のアプリケーションソフトウェアに固有のプロセスのためのプロセッサ（以下、システムに実装されているこれら汎用プロセッサ及び固有のプロセスのためのプロセッサを物理プロセッサとも呼ぶ）と、ステータス情報の内容に応じて、仮想的なプロセッサを汎用プロセッサと固有のプロセスのためのプロセッサとに切り替えて実行させる手段とを備えたことを特徴としている。

**【0006】** このマルチプロセッサ・システムでは、実装されたプロセッサとは別の「仮想プロセッサ」の概念を導入しており、システムにおいて実行すべきプロセスを、この仮想プロセッサに割り当てる。従って、アプリケーションソフトウェアのレベルでは、仮想プロセッサの使用を前提としたプログラムの記述を行えば足り、システムに実装された物理プロセッサの使用を前提としたプログラムの記述を行う必要はなくなる。

**【0007】** こうした仮想プロセッサの概念の導入によりアプリケーションソフトウェアのレベルでは物理プロセッサの使用を前提とする必要がなくなることを利用して、このマルチプロセッサ・システムでは、次のようにして、特定のアプリケーションソフトウェアに固有の拡張機能を、システムの上位互換性を保ったまま容易にシステムに追加できるようにしている。

**【0008】** 即ち、汎用プロセッサと、特定のアプリケーションソフトウェアに固有のプロセスのための特殊なプロセッサとの2通りの物理プロセッサを実装している。そして、仮想的なプロセッサに割り当てられた前記プロセスが特定のアプリケーションソフトウェアに固有のプロセスであるか否かを内容とするステータス情報を作成するようにしており、このステータス情報の内容に応じて、仮想的なプロセッサをこれらの汎用プロセッサと特殊なプロセッサとに切り替えて実行させるようにしている。

**【0009】** これにより、仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスではない一般的なプロセスであるときには、仮想プロセッサが汎用プロセッサで実行され、他方、仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスであるときには、仮想プロセッサがその固有のプロセスのためのプロセッサで実行されるようになる。

**【0010】** このように、このマルチプロセッサ・システムでは、仮想プロセッサの概念の導入を前提として、汎用プロセッサのアーキテクチャを全く変更しないことによりシステムの上位互換性を保ったまま、特定のアプリケーションソフトウェアに固有の拡張機能を容易にシステムに追加することができる。

**【0011】** しかも、システムを構成する際に、一般的

なプロセスのための汎用プロセッサの数と、アプリケーションソフトウェアに固有のプロセスのためのプロセッサの数とを、システムで実行させるアプリケーションソフトウェアに応じて決定することにより、効率的なシステム構成を実現することができる。

【0012】

【発明の実施の形態】以下、添付図面を参照して本発明の実施例を詳細に説明する。

《1. システムの構成例》図1は、本発明に係るマルチプロセッサ・システムの構成の一例を概念的に示す。このシステムには、OSやアプリケーションソフトウェア等を実行するためのプロセッサエレメントとして、2基の汎用のRISCプロセッサ1、2と、特定用途向けの（特定のアプリケーションソフトウェアに固有のプロセスのための）2基のSIMD（単一命令複数データ流方式）プロセッサ3、4とが設けられている。また、これらのプロセッサ1～4の他に、プロセッサ・マネージメント・ユニット（PMU）5が設けられている。PMU5は、仮想プロセッサを管理するための一種のコプロセッサである。

【0013】各プロセッサ1～4とPMU5とは、コミュニケーション・バスCBを介して接続されている。コミュニケーション・バスCBは、接続されている任意のエレメント間で多ビットの情報のやり取りが可能な双方向バスである。システムの主記憶装置であるシステムメモリSMも、このコミュニケーション・バスCBを介して各プロセッサ1～4に接続されることにより、各プロセッサ1～4に共有されている。

【0014】〈1a. PMUとPETLB〉このマルチプロセッサ・システムでは、仮想プロセッサと、物理プロセッサである各プロセッサ1～4とに、それぞれ固有のID（識別ラベル）が与えられている。仮想プロセッサのIDを「VEID」、物理プロセッサのうちRISCプロセッサ1、2のIDを「PEID」、物理プロセッサのうちSIMDプロセッサ3、4のIDを「SPEID（Special PEID）」と呼ぶ。

【0015】VEIDは、後出の〈4b. 仮想プロセッサの生成〉で述べるように、システムの起動後新たに仮想プロセッサを生成するときに、OSによってその仮想プロセッサに与えられたものである。PEIDとSPEIDは、システムの構成に応じてハードウェアによって与えられたものであり、変更することはできない。

【0016】PMU5（図1）内には、仮想プロセッサのVEIDと、その仮想プロセッサのステータス情報である「実行レベル」と、この仮想プロセッサへの割り当ての候補として定義される（この定義も、後出の〈4b. 仮想プロセッサの生成〉で述べるように、システムの起動後新たに仮想プロセッサを生成するときに、OSによって行われる）物理プロセッサ1～4のID（PEID及びSPEID）とを関連付けて記憶させることの

できる記憶領域（テーブル）が用意されている。このテーブルを、プロセッサエレメントテーブル（PETLB）と呼ぶ。

【0017】実行レベルは、通常のマикроプロセッサにおける特権レベルあるいは保護モードと同義のものか、あるいはそれを拡張させたものである。仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスではない一般的なプロセスである場合には、実行レベルはユーザー・レベルあるいはスーパーバイザ・レベルとなっている。他方、仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスである場合には、実行レベルはアプリケーション・レベルになっている。仮想プロセッサへのこの実行レベルの付与も、後出の〈4b. 仮想プロセッサの生成〉で述べるように、システムの起動後新たに仮想プロセッサを生成するときに、OSによって行われる。

【0018】図2は、プロセッサエレメントテーブル（PETLB）の一例を概念的に示す。PETLBは同図Aのように複数のエントリを持っており、各エントリ（但し図の最上部のシステムエントリSEを除く）は同図Bのようにいくつかのフィールドに分かれている。

【0019】これらのフィールドのうち、Validフィールド6は、そのエントリに有効なデータが存在することを示すためのフィールドであり、Lockフィールド7は、そのエントリの内容を変更することが禁止されていることを示すためのフィールドである。

【0020】同一エントリのLevelフィールド8、VEIDフィールド9、PEIDフィールド10、SPEIDフィールド11には、仮想プロセッサの実行レベル、その仮想プロセッサのVEID、その仮想プロセッサへの割り当ての候補として定義されたRISCプロセッサ1または2のPEID、その仮想プロセッサへの割り当ての候補として定義されたSIMDプロセッサ3または4のSPEIDが、それぞれ格納される。

【0021】各エントリに記憶される内容は、OSによってシステムメモリSM（図1）内で管理されている「プロセッサ・エレメント管理テーブル」（仮想プロセッサと物理プロセッサとの対応を定義するいわば仮想プロセッサの名簿）の一部分のコピーである。PETLBは一種の連想記憶装置になっており、VEIDを与えたとき、そのVEIDの仮想プロセッサに割り当てられるRISCプロセッサ1または2のPEIDやSIMDプロセッサ3または4のSPEIDを引き出せるようになっている。

【0022】尚、システムエントリSEは、後述のシステム・プロセッサに対する物理プロセッサPE2の割り当て情報を管理するための特別なエントリである。

【0023】PETLBのこうした構造は、一般的なマイクロプロセッサの仮想記憶方式に置いて用いられるテ

ーブルと似ている。

【0024】〈1b. システム・プロセッサ〉「システム・プロセッサ」は、仮想プロセッサの中の一つであり、システム全体に関わるエラーやイベントを処理するための特別な仮想プロセッサである。システム外部からの割り込みや、仮想プロセッサの管理に関するエラー等は、このシステム・プロセッサで発生したエラーとして処理される。

【0025】システム・プロセッサ以外の一般の仮想プロセッサについてのエントリの初期化は、通常はソフトウェアによってシステムメモリSM内のプロセッサ・エレメント管理テーブルで行わなければならないが、システム・プロセッサだけは、後出の〈3a. システムの初期化〉で述べるように、システムのリセット時にハードウェアによってPETLBでシステムエントリSEが初期化される。

【0026】また、PETLBのシステムエントリSE以外のエントリは、後出の《2. 仮想プロセッサへのアクセス》で述べるようにプロセッサ・エレメント管理テーブル中の他のエントリと入れ替えられることがあるが、システムエントリSEだけは、この入れ替えからは保護された位置にある。換言すれば、システム・プロセッサだけでは常に物理プロセッサが割り当てられており、システムエントリSEからその物理プロセッサを検索することができる。但し、システム・プロセッサにどの物理プロセッサを割り当てるは、システムエントリSE自体を書き換えることによっていつでも変更することが可能である。

【0027】《2. 仮想プロセッサへのアクセス》アプリケーションソフトウェアがシステム中のプロセッサにアクセスする場合、(例えば現在プロセスを処理中のプロセッサとは別のプロセッサにプロセスを投入したり、現在プロセスを処理中のプロセッサとは別のプロセッサと通信を行ったりするような場合等)には、VEIDを用いて仮想プロセッサに対してアクセスするようにする。換言すれば、アプリケーションソフトウェアでは、VEIDを用いて、仮想プロセッサの使用を前提としたプログラムの記述を行うようにする。

【0028】現在プロセスを処理中の物理プロセッサ(即ちプロセッサ1~4(図1)のうち現在実行中の仮想プロセッサに割り当てられた物理プロセッサ)の中で、このような別の仮想プロセッサに対するアクセスが発生すると、それが仮想プロセッサの呼び出しイベントとしてコミュニケーション・バスCB(図1)上に通知される。このとき、アクセスされた仮想プロセッサのVEIDも同時にバスCB上に乗せられる。

【0029】各物理プロセッサ1~4は、それぞれ実行中の仮想プロセッサのVEIDを記憶するための「VEIDレジスタ」を持っており、このレジスタを参照することにより、現在どの仮想プロセッサに割り当てられて

いるかを判別することができる。物理プロセッサは、バスCB上に乗せられたVEIDと現在割り当てられている仮想プロセッサのVEIDとが一致した場合(即ち自分が呼び出された仮想プロセッサである場合)、アクセスに対して直接応答する。

【0030】一方、PMU5(図1)は、PETLB(図2)の各エントリから、バスCBを介して送られたVEIDを格納しているVEIDフィールド9を検索する。そして、そのVEIDフィールド9と同じエントリ中から、当該仮想プロセッサに割り当てられる物理プロセッサのID(PEIDまたはSPEIDのいずれか)を得る(この部分の処理については、拡張機能と関連する処理なので、後出の《3. 拡張機能の使用方法》で再度詳述する)。そして、いずれの物理プロセッサ1~4もアクセスに対して直接応答しなかったときには、得られたIDを持つ物理プロセッサに対して、仮想プロセッサの入れ替えのための割り込み要求を発生する。

【0031】このIDを持つ物理プロセッサでは、この割り込みにより、VEIDレジスタ内のVEIDの書き換えと、例外ハンドラ(例外処理を記述したプログラム)によるプロセスの入れ替えが行われる。

【0032】尚、PMU5がPETLBの各エントリを検索した際に、バスCBを介して送られたVEIDを格納しているVEIDフィールド9がみつからなかった場合には、前述のシステム・プロセッサに例外が発生する。この場合には、OSが例外ハンドラによってPETLBのエントリの入れ替えを行い、その後改めて仮想プロセッサの呼び出しが行われる。

【0033】《3. 拡張機能の使用方法》《2. 仮想プロセッサへのアクセス》で述べたように、PMU5(図1)は、PETLB(図2)の各エントリから、バスCBを介して送られたVEIDを格納しているVEIDフィールド9を検索するが、そのVEIDフィールド9と同じエントリ中のLevelフィールド8の実行レベルがユーザー・レベルあるいはスーパーバイザ・レベルである場合(即ち、呼び出された仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスではない一般的なプロセスである場合)には、PMU5は、そのエントリ中のPEIDフィールド10からPEIDを得る。これにより、当該仮想プロセッサへの割り当ての候補として定義された物理プロセッサのうち、RISCプロセッサ1または2が実際に当該仮想プロセッサに割り当てられる。

【0034】これに対し、そのエントリ中のLevelフィールド8の実行レベルがアプリケーション・レベルである場合(即ち、呼び出された仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスである場合)には、PMU5は、そのエントリ中のSPEIDフィールド11からSPEIDを得る。これにより、当該仮想プロセッサへの割り

当ての候補として定義された物理プロセッサのうち、SIMDプロセッサ3または4が実際に当該仮想プロセッサに割り当てられる。

【0035】そして、《2. 仮想プロセッサへのアクセス》で述べたように、こうして割り当てられた物理プロセッサで、VEIDレジスタ内のVEIDの書き換えとプロセスの入れ替えとが行われる。

【0036】次に、現在実行中の仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスではない一般的なプロセスと特定のアプリケーションソフトウェアに固有のプロセスとの間で変化すると、当該仮想プロセッサを実行中の物理プロセッサが特定の命令を実行することにより、ユーザー・レベルあるいはスーパーバイザ・レベルとアプリケーション・レベルとの間での実行レベルの遷移が起こる。この実行レベルの遷移は、仮想プロセッサの呼び出しイベントとしてコミュニケーション・バスCB(図1)上に通知される。PMU5(図1)は、この呼び出しイベントで呼び出された仮想プロセッサについてのPETLB上のエントリ中のLevelフィールド8をこの遷移した実行レベルに更新した後、当該仮想プロセッサへの物理プロセッサの割り当てを行う。

【0037】この実行レベルの遷移時の物理プロセッサの割り当てには、次の3通りのパターンがある。

〈3a. 別の物理プロセッサへの割り当て(再割り当て)を伴わない実行レベルの遷移が起こった場合〉この場合には、それまで当該仮想プロセッサの実行が割り当てられていた物理プロセッサに、引き続き当該仮想プロセッサの実行を継続させる。

【0038】〈3b. 再割り当てを伴う実行レベルの遷移が起こり、且つ、新たに割り当てられた物理プロセッサがアイドル状態である場合〉この場合には、PMU5が、新たに割り当てられた物理プロセッサ(ユーザー・レベルあるいはスーパーバイザ・レベルに遷移した場合にはRISCプロセッサ1または2、アプリケーション・レベルに遷移した場合にはSIMDプロセッサ3または4)に対して、仮想プロセッサの入れ替えのための割り込み要求を発生することにより、その物理プロセッサに当該仮想プロセッサを実行させる。それまで当該仮想プロセッサの実行が割り当てられていた物理プロセッサは、アイドル状態に移行する。

【0039】〈3c. 再割り当てを伴う実行レベルの遷移が起こり、且つ、新たに割り当てられた物理プロセッサが他の仮想プロセッサを実行中である場合〉この場合には、それまで当該仮想プロセッサの実行が割り当てられていた物理プロセッサはアイドル状態に移行するが、新たに割り当てられた物理プロセッサでは、後述の〈4c. 仮想プロセッサの実行〉で述べるようにタイマ割り込みのような外部割り込みによってシステム全体のタスク切り替えが行われるまで、当該仮想プロセッサの実行

が開始されない。

【0040】《4. ソフトウェアとの連携による仮想プロセッサ管理の例》

〈4a. システムの初期化〉システムがリセットされると、各物理プロセッサ1~4は初期化されて待機状態になる。PETLB(図2)は、システムエントリSEが初期化され、それ以外のエントリが全て無効化される。

【0041】すべての初期化処理が終了すると、PMU5(図1)がシステム・プロセッサを呼び出す。この呼び出しは、《2. 仮想プロセッサへのアクセス》で述べた物理プロセッサに対する割り込み要求によって行われる。

【0042】起動したシステム・プロセッサでは、OSのカーネル(核)の実行が開始され、システムメモリSM(図1)内のプロセッサ・エレメント管理テーブルの初期化が行われる。

【0043】〈4b. 仮想プロセッサの生成〉その後、OSのその他の部分やアプリケーションソフトウェアが起動されて、それぞれのプロセスが新しい仮想プロセッサに割り当てられる。この処理は、プロセスが、OSの用意したプログラムである「仮想プロセッサ生成タスク」をシステム・コールで呼び出すことによって行われる。

【0044】呼び出された仮想プロセッサ生成タスクは、プロセスを割り当てられた仮想プロセッサにVEIDを与え、その仮想プロセッサに実行レベルを付与し(割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスではない一般的なプロセスであるときはユーザー・レベルあるいはスーパーバイザ・レベルを付加し、割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスであるときはアプリケーション・レベルを付与し)、その仮想プロセッサへの割り当ての候補となるRISCプロセッサ1または2とSIMDプロセッサ3または4とを定義する。そして、そのVEIDと実行レベルとPEIDとSPEIDと当該プロセスの実行に必要な情報とを、システムメモリSM(図1)内のプロセッサ・エレメント管理テーブルに記憶させる。このようにして、仮想プロセッサが生成される。

【0045】尚、新しく生成する仮想プロセッサへの割り当ての候補としてRISCプロセッサ1, 2のうちのいずれを割り当て、SIMDプロセッサ3, 4のうちのいずれを割り当てるかは、OSが各物理プロセッサ1~4の使用状況等をもとにして決定する。VEIDで表現可能な範囲内で、物理プロセッサの総数である4よりも多い数の仮想プロセッサを生成することも可能であり、その場合には複数の仮想プロセッサに同一の物理プロセッサを共有させるようにする。

【0046】〈4c. 仮想プロセッサの実行〉新しく生成された仮想プロセッサを実際に物理プロセッサ1~4

上で実行させる処理は、仮想プロセッサの生成処理とは独立かつ並行的に行われる。

【0047】そのために、システムには、タイマ割り込みのような外部割り込みが定期的にかかるようになっていいる。この仕組みは、単一のプロセッサに時分割によりマルチタスクを行わせるためのものと同じである。

【0048】タイマ割り込みはシステム全体に関係するイベントなので、それが発生すると、PMU5が物理プロセッサに対する割り込み要求によって自動的にシステム・プロセッサを呼び出す。この割り込み処理を記述したプログラムである割り込みハンドラから、OSの用意したプログラムである「仮想プロセッサ管理タスク」が呼び出されるようになっていいる。

【0049】呼び出された仮想プロセッサ管理タスクは、システムメモリSM内のプロセッサ・エレメント管理テーブルから、次のタイマ割り込みまでの一定期間物理プロセッサを割り当てて実行する仮想プロセッサを、各プロセス間の優先順位等をもとに決定する。こうして決定された仮想プロセッサが、《2. 仮想プロセッサへのアクセス》で述べたようアプリケーションソフトウェアによってアクセスされることにより、その仮想プロセッサの実行が開始される。

【0050】以上のように、このマルチプロセッサ・システムによれば、仮想プロセッサの導入を前提として、汎用の物理プロセッサであるRISCプロセッサ1、2のアーキテクチャを全く変更しないことによりシステムの上位互換性を保ったまま、特定のアプリケーションソフトウェアに固有の拡張機能を容易にシステムに追加することができる。

【0051】尚、以上の実施例では汎用の物理プロセッサであるRISCプロセッサと特定用途向けのSIMDプロセッサとをそれぞれ2基ずつ設けているが、これらのプロセッサをそれぞれ適宜の数だけ設けるようにしてよいことはもちろんである。特に、システムを構成する際に、システムで実行させるアプリケーションソフトウェアに応じてこれらのプロセッサの数を適宜決定することにより、効率的なシステム構成を実現することができる。

【0052】また、本発明が前提とするマルチプロセッサ・システムは、以上の実施例に示したものに限らず、

仮想プロセッサの概念を導入したシステムであれば適宜のものであってよい。

【0053】また、以上の実施例では、主記憶装置（システムメモリSM）を単一のバス（コミュニケーション・バスCB）を介して各プロセッサ（物理プロセッサ1～4）に共有させる単一バス方式の共有メモリ形（密結合形）マルチプロセッサに本発明を適用しているが、その他の方式（例えば多重バス方式、階層バス方式、マルチポートメモリ方式等）の共有メモリ形マルチプロセッサや、あるいはメッセージ交換形（疎結合形）マルチプロセッサに本発明を適用するようにしてもよい。

【0054】また、本発明は、以上の実施例に限らず、本発明の要旨を逸脱することなく、その他様々の構成をとりうることはもちろんである。

【0055】

【発明の効果】以上のように、本発明に係るマルチプロセッサ・システムによれば、汎用プロセッサのアーキテクチャを全く変更しないことによりシステムの上位互換性を保ったまま、特定のアプリケーションソフトウェアに固有の拡張機能を容易にシステムに追加することができる。

【0056】また、システムを構成する際に、一般的なプロセスのための汎用プロセッサの数と、アプリケーションソフトウェアに固有のプロセスのためのプロセッサの数とを、システムで実行させるアプリケーションソフトウェアに応じて決定することにより、効率的なシステム構成を実現することができる。

【図面の簡単な説明】

【図1】本発明に係るマルチプロセッサ・システムの構成の一例を示す図である。

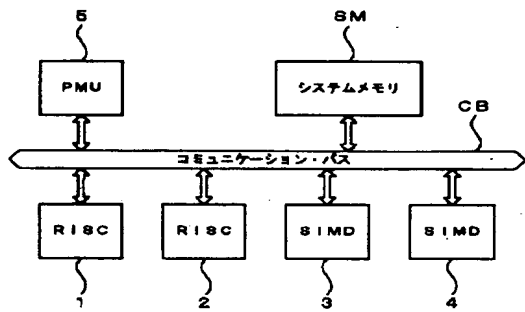
【図2】プロセッサエレメントテーブルの一例を示す図である。

【符号の説明】

1, 2 RISCプロセッサ、 3, 4 SIMDプロセッサ、 5 プロセッサ・マネージメント・ユニット、 CB コミュニケーション・バス、 SMシステムメモリ、 6 Validフィールド、 7 Lockフィールド、 8 Levelフィールド、 9 VEIDフィールド、 10 PEIDフィールド、 11 SPEIDフィールド



【図1】



【図2】

